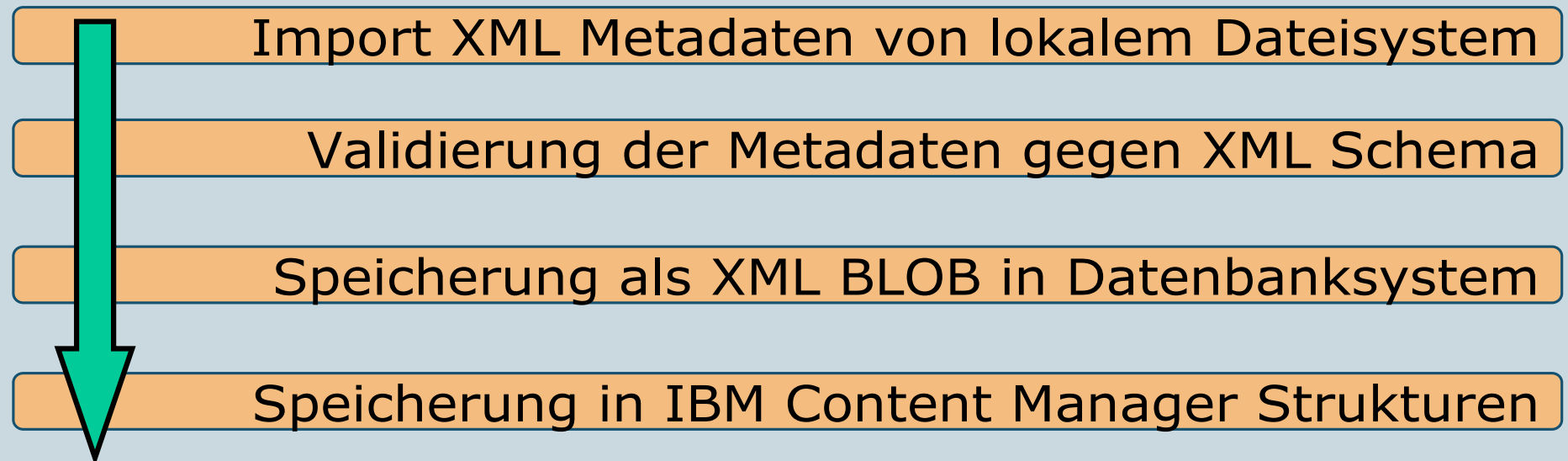


MyCoRe > V1.0: Technische Weiterentwicklung



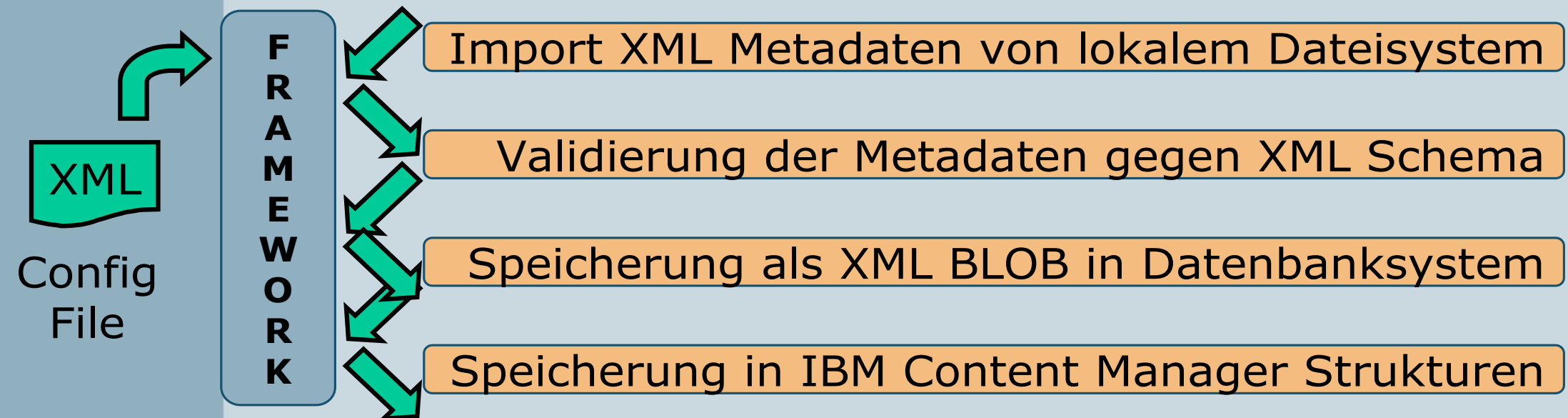
Kontrollflüsse bisher

- Kontrollflüsse sind im MyCoRe Kern oder der darauf basierenden Applikation fix und explizit definiert
- Content Store Implementierungen sind durch Konfiguration austauschbar
- aber geänderte Abläufe und Erweiterungen sind kaum ohne Programmierung oder Eingriff in den Kern möglich



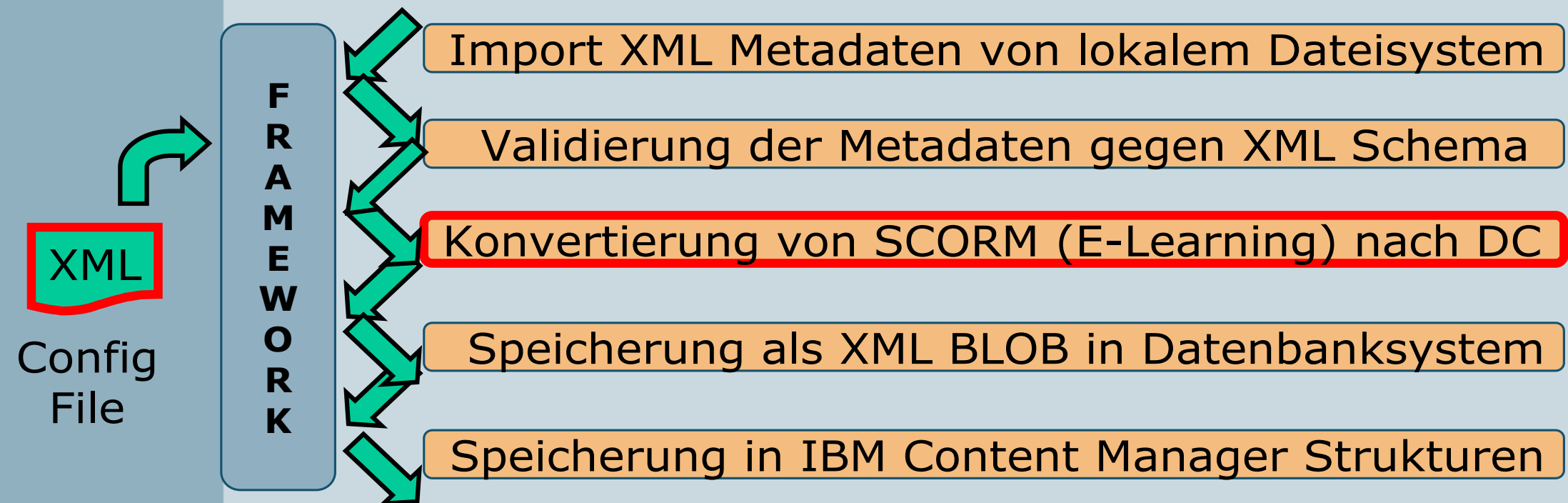
Kontrollflüsse künftig

- Unabhängige Komponenten realisieren Services
- *Inversion of Control*: das Framework steuert Abläufe und bildet eine „Pipeline“ von Komponenten
- Kontrollfluss wird konfiguriert, nicht programmiert
- Basis: Apache **Cocoon** und **Avalon** Framework
- Vorhandene Avalon Komponenten können genutzt werden, z. B. Email-Versand



Kontrollflüsse künftig

- Änderung der Abläufe oder Erweiterung durch Anpassen der zentralen Konfigurationsdatei
- Neue Applikationen können so flexibler aus MyCoRe Komponenten zusammengesetzt werden
- Über Wrapper-Komponenten sind externe Systeme integrierbar, z. B. Workflow Management



Typische Komponenten

- **Store:** Persistenz von Metadaten und Dateien
IBM CM 8.2, Filesystem, Streaming Server, Relationale Datenbank, XML-Datenbank
- **Indexer:** Daten für Suche indizieren
IBM CM 8.2, Jakarta Lucene Indizierung für Volltextsuche oder Suche in Metadaten
- **Extrahieren** von Metadaten aus Content
Anzahl Seiten aus PDF; Titel, Autor aus Word, EXIF-Informationen aus JPEG, Dauer hh:mm:ss aus Video,
- **Konvertieren** von Metadaten und Content
TIFF nach JPEG, SCORM nach Dublin Core, MPEG1 nach RealVideo9, Thumbnail aus JPEG
- **Importieren**
aus ZIP-Datei, aus Email-Attachment, per FTP, per WebDAV, von URL, aus OAIS SIP
- **Darstellen**
Video-Storyboard, Video mit Folien als SMIL Präsentation, Viewer für Rasterbilder

MyCoRe 2 Backends

- **Bisher:**
Direkte Verwendung von java.sql.* Klassen (**JDBC API**) und explizit codierten SQL Statements
-> Datenbankunabhängigkeit schwierig
z. B. MILESS: DB2, MySQL, Oracle
- **Künftig:**
Verwendung von **Hibernate** für objektrelationales Mapping
 - Unterstützung aller gängigen Datenbanken
 - Tabellen und SQL-Code automatisch generiert
 - Abbildung zwischen Java-Objekten und Tabellen automatisch

MyCoRe 2 Backends

- **Bisher:**
Direkte Verwendung von IBM CM Java API (Enterprise Information Portal)
 - Proprietäre Komponenten
- **Künftig:**
Verwendung von **JSR 170 API**
 - Standardisierte Java API für Content Repositories
 - Austauschbares CMS als MyCoRe Backend
 - Nutzung von IBM CM Funktionalität, ohne proprietär zu werden
 - Open Source Implementierung von JSR170

